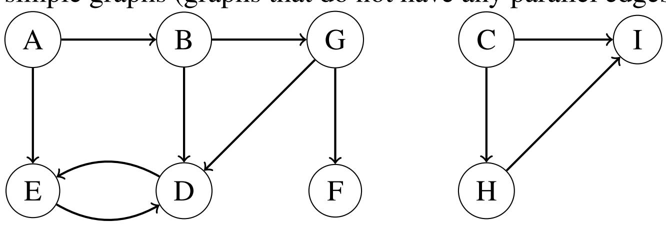
CMPSC 465 Data Structures & Algorithms
Spring 2024 Mehrdad Mahdavi and David Koslicki

Worksheet 5

Wednesday, February 14, 2024

1. **Graph Basics.** For parts (a) through (c), refer to the figure below. For parts (d) and (e), consider only simple graphs (graphs that do not have any parallel edges or self-loops).



Run DFS at node A, trying to visit nodes alphabetically (e.g. given a choice between nodes D and F, visit D first).

(a) List the nodes in the order you visit them (so each node should appear in the ordering exactly once).

(b) List each node with its pre- and post-number. The numbering starts from 1 and ends at 18.

(c) Label each edge as **T**ree, **B**ack, **F**orward or **C**ross.

(d) Let $|E|$ be the number of edges in a simple graph and $|V|$ be the number of vertices. Show that $|E| = O(|V|^2)$.

(e) For each vertex v_i in an undirected graph, let d_i be the *degree*- the number of edges incident to it. Show that $\sum d_i$ must be even.

2. **Pouring Water.** We have three containers whose sizes are 10 pints, 7 pints, and 4 pints, respectively. The 7-pint and 4-pint containers start out full of water, but the 10-pint container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 pints in the 7- or 4-pint container.

   1. Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered.
   2. What algorithm should be applied to solve the problem?
   3. Find the answer by applying the algorithm.

3. **Shortest Cycle.** Here's a proposal for how to find the length of the shortest cycle in an undirected graph with unit edge lengths.
When a back edge, say (v,w), is encountered during a depth-first search, it forms a cycle with the tree edges from w to v. The length of the cycle is $level[v] - level[w] + 1$, where the level of a vertex is its distance in the DFS tree from the root vertex. This suggests the following algorithm:

    1. Do a depth-first search, keeping track of the level of each vertex.
    2. Each time a back edge is encountered, compute the cycle length and save it if it is smaller than the shortest one previously seen.

    Show that this strategy does not always work by providing a counterexample as well as a brief (one- or two-sentence) explanation.

4. **Particular Edge Cycle.** Design a linear-time algorithm which, given an undirected graph G and a particular edge e in it, determines whether G has a cycle containing e.

5. **Post Number Ancestors.** Assume (u, v) is an edge in an undirected graph. Either prove the following statement or provide a counterexample: If during DFS $\texttt{post}(u) < \texttt{post}(v)$, then v is an ancestor of u in the DFS tree.